Kakas Bantu Analisis Kerancuan Kebutuhan Perangkat Lunak Berbasis Aturan

Yunata Dede Pratiwi, Daniel Oranova S dan Sarwosri
Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
Jalan Teknik Kimia, Kampus ITS Sukolilo, Surabaya 6011, Indonesia
Email: daniel@if.its.ac.id

***Abstrak*---Analisis kebutuhan adalah fase yang penting dalam pengembangan sebuah perangkat lunak. Otomatisasi evaluasi terhadap bahasa alamiah yang digunakan dalam dokumen kebutuhan telah ditetapkan untuk meningkatkan kualitas dari sebuah sistem sebelum memulai fase pembangunan sistem. Kakas Bantu Analisis Kerancuan Kebutuhan Perangkat Lunak Berbasis Aturan ini adalah suatu kakas bantu yang dapat membantu otomatisasi dalam evaluasi dokumen kebutuhan perangkat lunak. Kakas bantu ini mengekstraksi dokumen kebutuhan perangkat lunak menjadi kebutuhan spesifik. Kakas bantu melakukan analisis kerancuan pada kebutuhan spesifik dan memberikan rekomendasi atas kerancuan yang terjadi didalamnya.**

***Kata Kunci*— Analisis Kebutuhan, Kebutuhan Perangkat Lunak, Kerancuan.**

I. PENDAHULUAN

Tahap awal dari suatu pembangunan sistem perangkat lunak adalah proses analisis kebutuhan sistem. Di tahap awal ini terjadi interaksi antara calon pengguna sistem dan pihak pengembang perangkat lunak, dimana mereka harus memiliki persepsi yang sama atas spesifikasi perangkat lunak yang dibangun. Kumpulan dari spesifikasi kebutuhan dikumpulkan dalam bentuk Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL). Dokumen SKPL disusun menggunakan bahasa alamiah, yaitu bahasa yang kita gunakan dalam percakapan sehari-hari. Penggunaan bahasa alamiah memiliki kelemahan yaitu pada saat makna dari kalimat tersebut memiliki arti atau penafsiran yang berbeda-beda pada masing-masing orang. Dalam konteks ini, dokumen SKPL dapat ditafsirkan berbeda-beda oleh tim pengembang perangkat lunak. Dengan kata lain, dokumen SKPL tersebut bersifat rancu karena dapat memilki arti yang berbeda-beda. Dokumen SKPL yang bersifat rancu dapat menyebabkan kegagalan dalam pengembangan proyek perangkat lunak karena tujuan proyek tidak tercapai.

Kerancuan dalam sebuah dokumen SKPL disebabkan penggunaan kata yang rancu dalam menyusun kalimat-kalimat kebutuhan. Untuk itu pada tugas akhir ini dibangun sebuah kakas bantu analisis kerancuan kebutuhan perangkat lunak berbasis aturan. Kakas bantu ini dapat membantu tim pengembang perangkat lunak untuk mendeteksi kalimat rancu dalam kebutuhan spesifik sebuah dokumen SKPL beserta pemberian rekomendasi kepada pengguna terhadap kerancuan yang terjadi dalam kalimat tersebut. Kakas bantu ini hanya melayani analisis kerancuan pada dokumen SKPL berbahasa inggris, sesuai dengan format IEEE dan memiliki ekstensi berkas .docx.

II. KAJIAN PUSTAKA

A. Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak (*software requirements*) adalah atribut-atribut yang bersifat spesifik yang merupakan spesifikasi kebutuhan fungsional dan kebutuhan non fungsional dari sebuah sistem perangkat lunak.

Kebutuhan perangkat lunak dijabarkan dalam sebuah dokumen formal yang disebut dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL). Pada Tugas Akhir ini, bagian yang akan diekstraksi untuk melakukan analisis kerancuan adalah bagian "*Spesific Requirements*" atau Kebutuhan Spesifik dengan format tabel. Gambar 1 menunjukkan contoh format tabel kebutuhan spesifik yang akan diekstraksi.

3.1. Capability Requirements

General Capability Requirements

UID	Description	Priority
CAR001	The store is a separate part of the system. A copy of the store qualifies as a backup.	1
CAR002	The system gives a visual indication to the user of the kind of actions it is performing, and provides a progress indication of search, upload and download operations.	1

Gambar 1. Format Tabel Kebutuhan Spesifik

Bagian yang diambil untuk diekstraksi adalah pada bagian kolom 2 tabel *spesific requirements* dimana dalam kolom tersebut terdapat daftar pernyataan kebutuhan.

B. Kerancuan Kebutuhan Perangkat Lunak

Kerancuan dalam sebuah dokumen SKPL dapat terjadi dalam berbagai bentuk bahasa alamiah. Misalnya saja kita jumpai sebuah pernyataan:

"Design a program that allows a network operator to plan changes in every parameter of every cell in the network. These planned changes should be verified for consistency and correctness and then applied to the network in the least disturbing way"[1].

Dalam pernyataan diatas, tidak dijelaskan bagaimana sebuah sistem memverifikasi sebuah konsistensi dan kebenaran lalu mengimplementasikannya dalam jaringan tersebut. Sehingga pernyataan ini tidak dapat diimpementasikan dalam rancangan sistem apalagi dibangun menjadi sebuah fungsi dari sistem.

C. *Analisis Kerancuan Kebutuhan Perangkat Lunak Berbasis Aturan*

Aturan ini dibuat berdasarkan pola-pola kalimat yang dirumuskan berdasarkan aturan SMART Requirement yang dikerjakan pada penelitian sebelumnya[2]. Aturan ini dibangun untuk mendeteksi kerancuan dalam sebuah pernyataan spesifikasi kebutuhan perangkat lunak. Aturan ini mengacu pada sebuah aturan SMART *Requirements* dengan menggunakan pengolahan bahasa alamiah yang dibantu oleh WordNet. SMART *Requirements* adalah sebuah akronim yang merepresentasikan lima buah acuan yang menentukan kualitas kebutuhan perangkat lunak. Lima acuan tersebut adalah Spesific, Measurable, Attainable, Realisable dan Traceable[3].

Bagian yang menjadi fokus utama dalam Tugas Akhir ini adalah bagian pertama yaitu *spesific*. Bagian *spesific* menjelaskan bahwa sebuah sebuah kebutuhan perangkat lunak tidak bersifat rancu.

III. METODOLOGI

A. *Analisis*

Pada proses analisis akan dijelaskan mengenai deskripsi umum dari kakas bantu dan arsitektur sistem dari kakas bantu.

1) *Deskripsi Umum*

Dalam Tugas Akhir ini dibuat suatu aplikasi yang mampu melakukan analisis kerancuan kebutuhan perangkat lunak dari sebuah dokumen SKPL. Aplikasi ini mampu melakukan ekstraksi terhadap kebutuhan spesifik yang terdapat dalam sebuah dokumen perangkat lunak dan mampu meberikan rekomendasi atas kerancuan yang terjadi pada pernyataan kebutuhan di dalam sebuah kebutuhan spesifik.

2) *Arsitektur Sistem*

Sistem ini hanya dapat diakses oleh pengguna dimana sistem terinstalasi. Pengguna dalam kasus ini adalah seorang analis sistem kebutuhan perangkat lunak. Pengguna akan memasukan dokumen SKPL sebagai masukan sebagai sistem untuk mengetahui kerancuan dari dokumen SKPL tersebut.

Dokumen SKPL akan diekstraksi menggunakan bantuan dari pustaka Apache POI. Setelah dilakukan ekstraksi dokumen SKPL, sistem akan mengambil isi dari kebutuhan spesifik. Lalu akan dilanjutkan dengan pengecekan kalimat oleh pustaka openNLP. Setelah itu, kalimat yang berhasil lolos sebagai kalimat utuh, akan mengalami proses pemisahan kalimat berdasarkan strukturnya menggunakan pustaka *Stanford Pos Tagger* . Setelah kalimat terpecah berdasarkan POS nya, maka akan dilakukan analisis kerancuan dengan menggunakan aturan yang terdaftar dalam sebuah tabel aturan dalam basis data. Sehingga akan diperoleh hasil apakah kalimat tersebut rancu atau tidak. Sehingga dapat ditentukan rekomendasi yang akan diberikan. Dan hasil output akan dicetak dan ditampilkan pada pengguna.

B. *Perancangan*

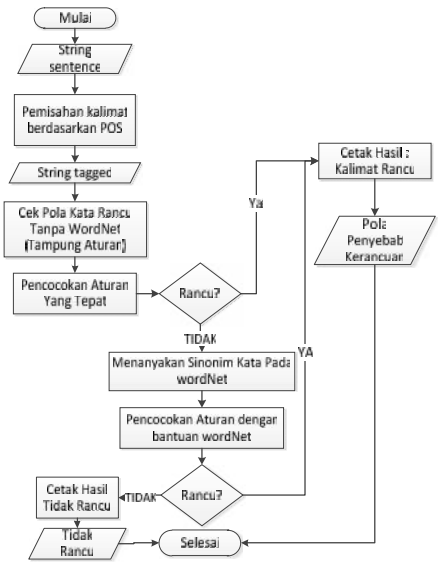
Proses utama dalam kakas bantu dibagi menjadi 3 buah proses dimana masukan dan keluaran dari masing-masing proses berhubungan satu dengan lainnya.

1) *Proses Ekstraksi Dokumen*

Masukan dokumen SKPL yang diproses adalah dokumen dengan ekstensi .docx. Ekstraksi dokumen dilakukan pada dokumen SKPL denga format Microsoft Word (.docx) menjadi kebutuhan spesifik. Kebutuhan spesifik dokumen SKPL diperoleh dengan menggunakan pustaka apache POI. Id dari kebutuhan spesifik dan format tabel digunakan sebagai penanda kebutuhan spesifik yang akan diekstraksi. Setelah melalui proses ekstraksi, akan dilakukan proses pengecekan kalimat utuh dengan menggunakan pustaka OpenNLP. Pengecekan kalimat utuh dilakukan agar hanya kalimat bahasa Inggris yang utuh yang dapat masuk dalam proses analisis kerancuan.

2) *Proses Analisis Kerancuan*

Pemisahan kalimat berdasaran struktur kalimatnya atau POS nya dilakukan dengan menggunakan pustaka `stanford pos taggers`. Digunakan juga pustaka wordNet untuk membantu proses analisis kerancuan dengan menggunakan sinonim. Gambar 2 menunjukkan diagram alir proses analisis kerancuan.



Gambar 2. Diagram Alir Analisis Kerancuan

Pencocokkan aturan yang tepat dilakukan pada tabel *aturansmart* pada basis data mySql. Proses analisis dimulai dengan mencocokkan masing-masing kata dalam kalimat dengan kata-kata rancu dalam tabel aturan. Jika terjadi kecocokan salah satu kata dengan kata yang tersimpan dalam tabel aturan, maka kalimat tersebut dinilai rancu.

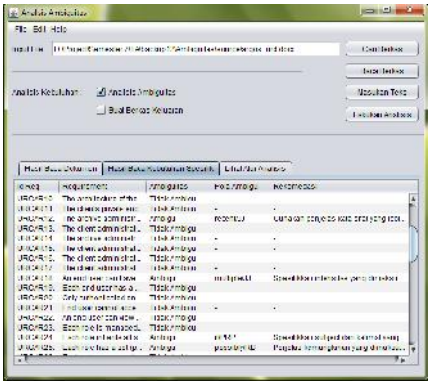
Saat tidak ditemukan kata rancu yang persis sama dengan kata-kata rancu pada tabel aturan, dilakukan penyimpanan pos kalimat yang memiliki pos sama dengan pos kata rancu. Disinilah pustaka wordNet bekerja untuk mencari sinonim dari kata-kata rancu yang terdapat dalam tabel aturan. Pencarian sinonim hanya dilakukan pada kata yang memiliki pos sama dengan kata yang sedang diperiksa. Apabila sinonim ditemukan, maka kalimat tersebut akan dinyatakan rancu.

3) Proses Pemberian Rekomendasi

Rekomendasi hanya diberikan pada kalimat yang memiliki hasil analisis berupa kalimat rancu. Pemberian rekomendasi dilakukan dengan mencocokkan pola penyebab rancu dalam tabel basis data aturan dengan rekomendasi yang terdapat di dalamnya. Pada tabel basis data aturan pola penyebab rancu dan rekomendasinya tersimpan dalam satu baris yang sama. Ketika kerancuan disebabkan oleh penggunaan sinonim kata rancu, maka akan dilakukan pengecekan ulang terhadap pola kata penyebab rancu. Hal tersebut dilakukan untuk mendapatkan referensi pola penyebab kerancuan sehingga didapatkanlah hasil rekomendasinya.

IV. UJI COBA DAN EVALUASI

Uji coba dilakukan dengan menguji keseluruhan fungsionalitas dalam kakas bantu yang berupa, fungsi ekstraksi dokumen SKPL, analisis kerancuan kebutuhan spesifik dan pemberian rekomendasi atas kerancuan yang terjadi. Gambar 3 menunjukkan antarmuka kakas bantu pada saat melakukan pengujian fungsional dengan menggunakan dokumen argos_urd.docx sebagai data uji.



Gambar 3. Antarmuka Kakas Bantu Saat Melakukan Pengujian Fungsional

Setelah melakukan uji coba fungsional, dilakukan pengujian akurasi performa hasil analisis. Uji coba ini bertujuan untuk mengetahui tingkat akurasi/ kebenaran dari analisis kerancuan yang dihasilkan sistem dengan mencari indeks Kappa(indeks kesepakatan anatara penguji 1 dan penguji 2). Hasil perhitungan manual hasil yang didapat dari uji coba ini terdapat pada Tabel 1.

Tabel 1.
Hasil Perhitungan Hasil Analisis Kerancuan

		Kakas Bantu	
		Rancu	Tidak Rancu
Ahli	Rancu	42	0
	Tidak Rancu	8	15

Dimana penguji 1 adalah seorang ahli kebutuhan perangkat lunak dan penguji 2 adalah kakas bantu analisis kerancuan berbasis aturan. Dokumen yang digunakan dalam uji coba ini adalah dokumen uji dari penelitian ReqSac[4]. Dimana pernyataan kebutuhan dalam dokumen tersebut telah diklasifikasikan sebelumnya oleh ahli kebutuhan perangkat lunak.

Berdasarkan data tersebut maka P(A) = (42 + 16) / 65 = 0.8923. Kemudian untuk menghitung P(E) dilakukan dengan cara sebagai berikut:

1. Ahli menjawab rancu sebanyak 42 kali dan tidak rancu sebanyak 23 kali, maka prosentase ahli menjawab rancu = 64.61 %
2. Kakas bantu menjawab rancu sebanyak 49 kali dan tidak rancu sebanyak 16 kali, maka prosentase kakas bantu menjawab rancu adalah sebesar = 75.38 %
3. Probabilitas para penguji menjawab rancu adalah 0.6461 * 0.7538 = 0.4870
4. Probabilitas para penguji menjawab tidak rancu adalah (0.3539 * 0.2461) = 0.0870
5. P(E) = 0.4870 + 0.0870 = 0.5740

Setelah itu nilai κ (indeks Kappa) dapat ditentukan dengan:

κ = (0.8923 – 0.5740) / (1-0.5740)
= **0.3183/0.426**
= 0.7471

V. KESIMPULAN

Kesimpulan yang didapat dari Tugas Akhir ini yaitu:

1. Kakas bantu dapat melakukan ekstraksi terhadap sebuah dokumen SKPL dengan ekstensi dokumen docx menjadi sebuah kebutuhan spesifik. Analisis kerancuan dan pemberian rekomendasi berhasil dilakukan pada kebutuhan spesifik yang berhasil diekstraksi.
2. Indeks kappa yang didapatkan dari hasil dokumen uji penelitian ReqSac didapatkan hasil kappa sebesar 0,7079 yang artinya kakas bantu memiliki nilai proporsi kesepakatan Banyak *(Substantial).*

UCAPAN TERIMA KASIH

Penulis Y.D.P mengucapkan terima kasih kepada kedua orangtua dan keluarga penulis, dosen pembimbing, dosen dan kepala jurusan Teknik Informatika, kerabat-kerabat dekat, serta berbagai pihak yang telah membantu penulis dalam menyelesaikan Tugas Akhir ini.

DAFTAR PUSTAKA

[1] Wilson, W., Rosenberg, L., & Hyatt, L. "Automated Quality Analysis of Natural Language Requirement Specifications". Proceedings 14th Annual Pacific Northwest Software Quality Conference, ortland, (1996).

[2] Muliawan, I Wayan, "Analisis Ambiguitas Kebutuhan Perangkat Lunak berdasarkan Acuan Smart Requirement". Tesis di Jurusan Teknik Informatika, Institut Teknologi Sepuluh Nopember, Surabaya, (2011).

[3] Mannion, M., Keepence, B. "SMART Requirements". ACM Sigsoft. The Standish Group International, Inc. "The CHAOS Report", (1995).

[4] Hussain, H. M. I., "Using Text Classification to Automate Ambiguity Detection in SRS Documents". Tesis di Departemen Computer Science & Software Engineering, Universitas Concordia , Canada, (2007).